

Benutzerhandbuch

Motorsteuerkarte MSK351

Version 2.0

Lizenzbestimmungen

Die in diesem Handbuch beschriebene Software wird Ihnen gemäß den Bedingungen eines Lizenzabkommens zur Verfügung gestellt und darf nur unter den darin beschriebenen Bedingungen eingesetzt werden.

Daten, Abbildungen, Änderungen

Daten und Abbildungen sind unverbindlich. Änderungen, die dem technischen Fortschritt dienen, sind vorbehalten. Falls Sie Verbesserungs– oder Änderungsvorschläge haben oder Fehler in dieser Druckschrift entdecken sollten, bitten wir um Ihre Mitteilung.

Gewährleistung

KEINE GEWÄHRLEISTUNG. Die technische Dokumentation wird Ihnen OHNE Mängelgewähr geliefert. OHP übernimmt keine Gewährleistung für deren Genauigkeit oder Verwendung. Die Verwendung der technischen Dokumentation oder der darin enthaltenen Informationen hat der Benutzer zu verantworten.

Schulung

Zur Vermittlung ergänzender Systemkenntnisse werden von OHP entsprechende Schulungen angeboten.

Warenzeichen

OHP, das OHP-Logo und ProWin sind Marken der OHP GmbH.

Die übrigen in diesem Handbuch verwendeten Produktnamen können eingetragene Warenzeichen und/oder Warenzeichen der jeweiligen Unternehmen sein.

Copyright



Inhalt



Teil I Allgemeines

1 Gültigkeitshinweis

Diese Beschreibung gilt für die Motorsteuerkarte MSK351und MULTIPROG Version 4.8. Die aktuelle Firmwareversion stellen wir auf www.ohp.de im Kundenbereich als Download zur Verfügung.

Teilleistungen werden möglicherweise von älteren Firmwareversionen nicht unterstützt. Eine Aufstellung der Firmwareversionen und die zugehörigen neuen Leistungen finden Sie in **TEIL II Projektierung und Programmierung,** Kapitel 1.

2 Einführung

Die MSK351 versteht sich als intelligente Motorsteuerkarte die autark oder in einem @120 Baugruppenträger in Verbindung mit einer ALU320, einer DEA 300, einem Standard PC mit Multiprog oder ProWin betrieben werden kann.

Die MSK kann als eigenständige Klein SPS betrieben werden, die mit der IEC 61131 kompatiblen Programmiersprache Multiprog programmiert wird. Zur Kopplung steht eine Ethernet Schnittstelle zur Verfügung, die gleichzeitig auch als Programmierschnittstelle dient.

Die MSK verfügt über 16 Digitale Eingänge, 12 Digital Ausgänge sowie einen analogen Eingang
mit 0-20mA. Das integrierte Netzteil der MSK351 wird mit 24 V betrieben.

3 Systemvoraussetzungen

3.1 Systemvoraussetzungen Programmiersoftware

Hardware

Für die optimale Leistungsfähigkeit des SPS-Programmiersystems MULTIPROG müssen mindestens die folgenden Hardware-Voraussetzungen erfüllt sein:

Gerät/Baugruppe	Minimal	Empfohlen ab
IBM-kompatibler PC mit Pentium-Prozessor	Pentium II 350 MHz	Pentium III 500 MHz
Arbeitsspeicher	64 MB	128 MB
Festplatte	250 MB freier Speicherplatz	
CD-ROM-Laufwerk	benötigt	
VGA-Monitor Farbeinstellungen Auflösung	 256 Farben 800 x 600	 True Color 1024 x 768
Ethernet-Schnittstelle	benötigt	

Software

MULTIPROG läuft unter folgenden Betriebssystemen:

- Microsoft Windows XP, SP2
- Microsoft Windows Vista
- Microsoft Windows 7



4 Kommunikationsschnittstellen der MSK351

Die MSK351 verfügt über je eine Ethernet Schnittstelle (RJ45) und eine RS232 Schnittstelle (D-Sub9), die auf der Modulunterseite sind.

Auf der Rückseite des Moduls befindet sich ein PAB Stecker mit dem das Modul an den PAB Bus der ALU320 (DEA300) angeschlossen werden kann. Die Kopplung der Ethernet-Schnittstelle erfolgt über Modbus TCP, d.h. der Datenaustausch kann mit einer @120 einer @250 oder mit einem Standard Modbus TCP Client erfolgen.

Die Verbindung zum Programmiergerät erfolgt ebenfalls über Ethernet. Das bedeutet, dass bei Kommunikation über TCP/IP ein Switch o.Ä. vorgeschaltet werden muss, damit gleichzeitig das Programmiergerät angeschlossen werden kann.

4.1 MSK351 Kommunikationsprozeduren

Die folgenden Kommunikationsprozeduren können direkt auf der MSK betrieben werden können. Verfügbar sind fernwirktechnische Protokolle und Feldbusprotokolle, wie in der folgenden Tabelle aufgeführt. Die Projektierung erfolgt je nach Ausführung der MSK351 und Prozedur entweder mit MULTIPROG oder mit einer Parameterdatei.

Prozedur	Modul	Anschluss	Betriebsart	Anzahl Schnittst.	Projektierung	Übertragungsbaugruppen
SEAB-1F	MSK351	RS232	Slave	1	Multiprog, Anwenderprogramm	UEM002, -202
IEC 60870-5-101	MSK351	RS232	Slave	1	Multiprog, Parameterdatei	UEM002, -202
IEC 60870-5-101 AWD	MSK351	RS232	Slave	1	Multiprog, Parameterdatei	UEM300, -301, -302, -304, -004, -005,-007
IEC 60870-5-103	MSK351	RS232	Slave	1	Parameterdatei	
IEC 60870-5-104	MSK351	Ethernet	Server	1	Multiprog, Parameterdatei	UEM306, -007
Modbus TCP	MSK351	Ethernet	Server	4	Multiprog, Parameterdatei	UEM306, -310
Modbus RTU	MSK351	RS232	Slave	1	Multiprog, Parameterdatei	UEM002, -202

4.2 Ankopplung über Modbus

Modbus RS232 Slave / Modbus TCP-Server

Die serielle Schnittstelle der MSK kann auch als Modbus Slave Schnittstelle projektiert werden. Der Modbus TCP-Server wird über die Ethernet-Schnittstelle angekoppelt.

Über diese Schnittstellen können beliebige Modbus-Master / Modbus-Clients Daten aus der Station abfragen.

Bis auf die Auswahl des Interfaces (SERIELL oder TCP_IP) ist die Projektierung für beide Protokolle identisch.

Vom der MSK werden folgende Funktionscodes unterstützt:

FC 1	Lese Bit	Read Coils
FC 2	Lese Eingang	Read Discrete Inputs
FC 3	Lese Register	Read Holding Register
FC 4	Lese Eingangswort	Read Input Register
FC 5	Schreibe einzelnes Bit	Write Single Coil
FC 6	Schreibe einzelnes Register	Write Single Register
FC 15	Schreibe Bitbereich	Write Multiple Coils
FC 16	Schreibe Registerbereich	Write Multiple Register
FC 23	Lese/Schreibe Registerbereich	Read/Write Multiple Register

Die Daten werden im Shared Memory Bereich zur Verfügung gestellt. Die Projektierung erfolgt über INI-Dateien, die auf die Micro-SD Karte gespeichert werden.

Anmerkung: Die Funktionen FC 2 und FC 4 greifen nicht direkt auf die Eingänge der Hardware zu. Die Eingänge müssen auch hier auf dem Shared Memory Bereich abgebildet werden.



Hinweis: Für die Kopplung zu einer @120 oder @250 mit Modbus-TCP ist zu beachten, dass pro MSK351 zwei Modbus-TCP Verbindungen (eine in Schreib- und eine in Leserichtung) benötigt werden. Eine @120/250 kann maximal 100 Verbindungen verwalten. Dies bedeutet, dass bei direkter Kopplung einer MSK351 mit einer @120/250, maximal 50 MSK351 angekoppelt werden können. Sollen mehr über den Modbus-TCP betrieben werden, so sind die Baugruppen in einem Magazin (DTA) mit einer ALU320 oder DEA300 zu betreiben. Pro Station können so über das Magazin insgesamt 19 MSK351 betrieben werden, wobei 2 Verbindungen pro ALU/DEA erforderlich sind. In dieser Betriebsart können > 120 MSK351 an einer übergeordneten Station betrieben werden. Ein Mischbetrieb zwischen der Stand-Alone-Variante und Betrieb mit ALU/DEA ist zulässig.



Weiterführende Informationen siehe **Teil II Projektierung und Programmierung** Kapitel 5.2, Kapitel 5.3 und Kapitel 5.5.



4.3 Zugriff auf die Micro SD-Karte über ActiveSync

Die MSK351 besitzt einen Micro-USB-Anschluss, um über diesen mit ActiveSync auf einfache Weise auf die interne Micro-SD Karte zugreifen zu können. Um diese Funktion nutzen zu können, muss auf dem PC Active-Sync von Microsoft installiert sein. Dieses Tool kann kostenfrei aus dem Internet herunter geladen werden.

Wird nun per USB-Kabel eine Verbindung zwischen MSK351 und PC hergestellt, so wird nach kurzer Zeit ActiveSync aktiv. Dort ist dann die Option „Nein, keine Partnerschaft festlegen“ zu wählen.



Anschließend kann dann über „Arbeitsplatz“ → „Mobiles Gerät“ auf die SD-Karte zugegriffen werden.



TEIL II Projektierung und Programmierung

1 Neue Leistungen der MSK351 Firmware

Version 3.00

Serienfreigabe

Version 3.10

- Einsatz von PROCONOS4 für Funktion Online-Ändern
- IEC101 AWD - Initstrings angepasst
- IEC101 AWD - Zeitüberwachung (TOUT) angepasst

Version 3.25

- Modbus TCP/IP und Seriell eingebunden

Version 3.30

- Systemzeit jetzt mit Millisekunden-Auflösung.
- Wegen Overload-Performance den Modbus TCP/IP auf 6 Request pro Server-Aufruf begrenzt.

Version 3.33

- Neue ProConOS Version eingebunden.
- SendCom.exe V1.0 Initialisierungshilfsprogramm für Modems im Verzeichnis. "sd card\wcs_sys"

Version 3.35

- Treiber für IEC103 eingebunden (Modul mit IEC101 Datentypen im privaten Bereich der IEC103)
- IEC60870-5-101: Zeitstempel Aktualisierung für SP_TM, DP_TM, BO_TM korrigiert.
- IEC60870-5-104: Zeitstempel Aktualisierung für SP_TM, DP_TM, BO_TM korrigiert.

Version 3.37

- IEC103 Modul mit IEC101 Datentypen SP, DP, BO ohne Zeitstempel aktiviert
- IEC103 Ursache Erstanlauf/Wiederanlauf korrigiert!

Version 3.42

- Neue Firmwarebibliotheken eingebunden

  ALUX20_OHP4_SYS_OS
  Die Firmware-Bibliothek stellt für die Micro A020 mehrere Bausteine für die Steuerung des RASVPN Dienstes und ein Reboot Kommando für die ALU zur Verfügung.

AX20_SOS_REBOOT ... Neustart(Warmstart) der ALU
AX20_SOS_RAS_DISC ... Verbindungsabbau RASVPN Dienst ausführen
AX20_SOS_RASVPN_STATUS ... Verbindungsstatus und Betriebszustand des RASVPN Dienst abfragen
AX20_SOS_RAS_MODEMRESET ... HW-Reset (Kaltstart) für internes MODEM ausführen

ALU020_OHP4_AWPWDG
Watchdog über das Anwenderprogramm zur Überwachung einer Anwendertask (z.B. default).

- Systemmerker erweitert
  Systemmerker %MD1.3380: Breakpoint Aktiv, 0 ... kein Breakpoint, 1 ... Breakpoint vorhanden
  Systemmerker %MD1.3384: HW Watchdog Status,
  0x00000000 = Watchdog nicht aktiv
  0x00000001 = Watchdog aktiv
  0x00000002 = Watchdog inaktiv - Breakpoint gesetzt
  0x00000004 = Watchdog inaktiv - DL Changes , etc
  0xFFFFFF00 = Watchdog ausgelöst
- Modbus TCP/IP:
  Durch Netzwerkstörungen (WLAN/VPN) offene Verbindungen (inaktiver Port) werden jetzt geschlossen!
  Eine Überprüfung erfolgt über eine 3 stufige Inaktivitätsüberwachung für die TCP-Kommunikation.
  Der Server schließt die TCP-Verbindung sobald er innerhalb der Inaktivitätsphasen keine TCP-Anfragen mehr erhält.

  1. Phase: Jede Verbindung wird auf max. Inaktivität von 12 Stunden überprüft.
  2. Phase: Bei Verbindungsaufbau werden alle Ports mit der gleichen IP-Adresse auf 2 Minuten Inaktivität geprüft.
  3. Phase: Ist die max. Anzahl (CMAX_MODBUS_TCPIP_CONN = 4) von Verbindungen erreicht, dann werden alle IP Verbindungen auf 99 Sekunden Inaktivität geprüft.

Version 3.43

- Benutzerautorisierung für die Dienste FTP und TELNET über Datei ALUUSER.INI.

Version 3.44

- SEAB-1F Slave Firmware-Bibliothek eingebunden.
- SEAB-1N Slave Firmware-Bibliothek eingebunden



2 Erzeugen eines neuen Projektes in MULTIPROG

Beim Einrichten eines neuen Projekts gehen Sie folgendermaßen vor:

Schritt 1 Im Menü Datei "Neues Projekt…" anwählen.

Schritt 2 Vorlage MSK351 mit Doppelklick auswählen.

Schritt 3 Gegebenenfalls die IP-Adresse für die Verbindung zur MSK ändern (Teil II, Kapitel 3).

Schritt 4 Projekt über Menü Datei "Projekt speichern unter…" mit Projektnamen abspeichern.

Schritt 5 Wenn die Kommunikationsprotokolle IEC 60870-5-101 oder -104 verwendet werden, mit Hilfe der OHP-Toolbar die Projektierung erzeugen.

Schritt 6 Logische Programm-Organisationseinheiten (POE) erstellen oder das automatisch erzeugte Standard-AWP erweitern.

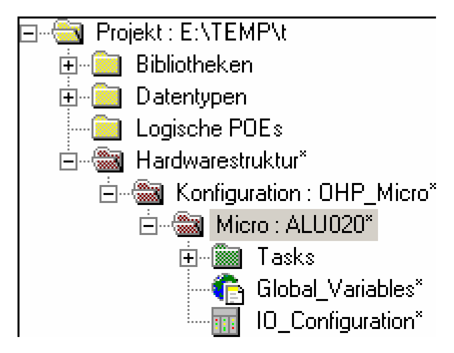
Schritt 7 Die erstellten POEs in die Liste der auszuführenden Tasks eintragen.

Schritt 8 Programmcode erzeugen (F9 oder Menü "Code -> Make").

Schritt 9 Programmcode über das Menü "Online -> Projektkontrolle" in die MSK senden und starten.

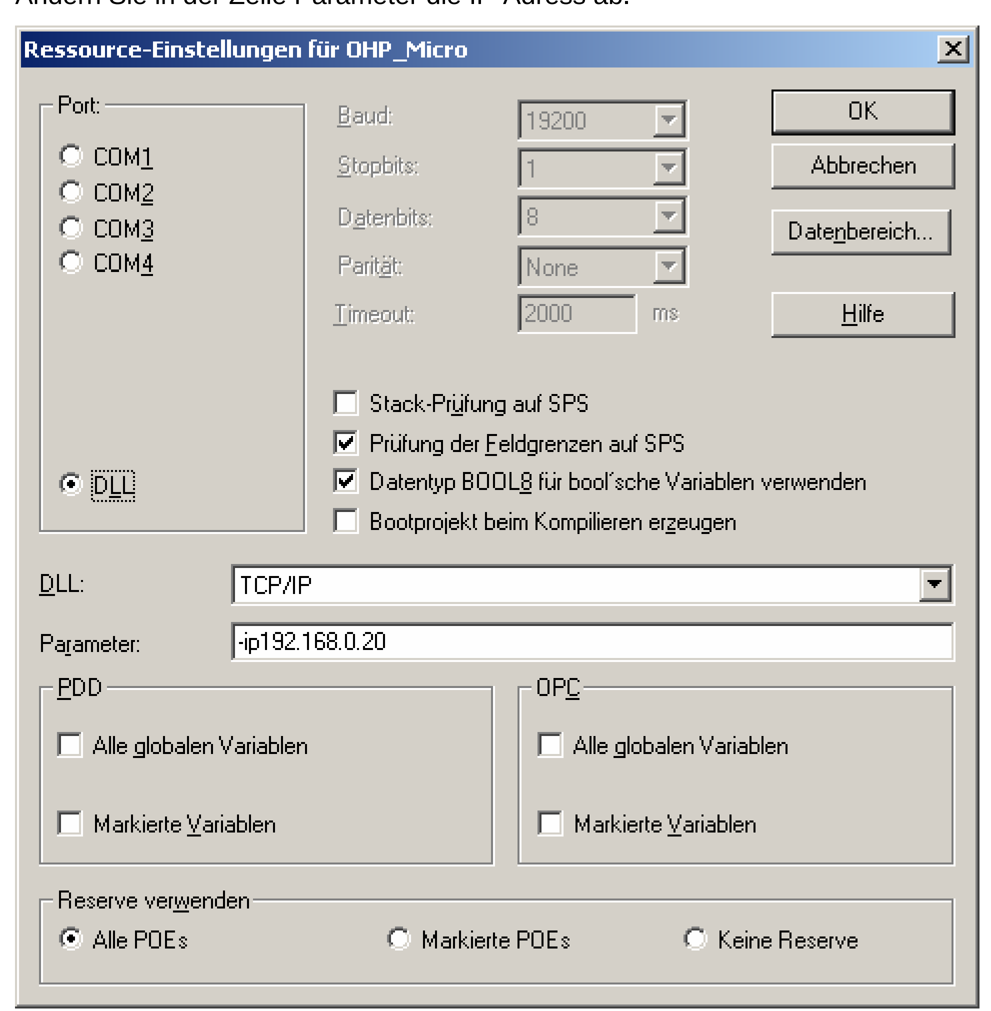


3 Ändern der IP-Adresse für die Verbindung zur MSK



Wählen Sie in der Hardwarekonfiguration die Ressource, im Beispiel Micro:A020, mit der rechten Maustaste an. Öffnen Sie über das Pop-UP-Fenster den Dialog "Einstellungen… ".

Ändern Sie in der Zeile Parameter die IP-Adress ab.





4 Definition der Ein- und Ausgaben

Bei Auslieferung ist die MSK351 bereits mit einem Standardanwenderprogramm ausgerüstet, welches die Hardware Ein- und Ausgänge mit den DPM-Übergabeadressen verbindet. Dadurch kann die MSK351 als einfaches E/A-Modul direkt zusammen mit ALU320 oder DEA300 betrieben werden.

Der Analogwert (0-20mA = 0…4095) wird über das Standardprogramm auf den Wertebereich 0…32000 skaliert, somit ist die MSK351 im Bereich kompatibel zum ADU206/306/308.

Die Programmquellen sind auf der SD-Karte abgelegt und können mit MP@Plus hochgeladen werden. Dieses Standardprogramm kann mit MP@Plus erweitert werden, um spezielle Funktionen zu realisieren.

4.1 Datenkopplung von / zur DEA300 oder ALU320

Die MSK351 besitzt eine DPM-Schnittstelle zur DEA300 bzw. ALU320, über welche Daten ausgetauscht werden können. Diese Schnittstelle bildet sich als virtuelle I/O Karten ab, die in Multiprog wie gewöhnliche I/O Karten (DEP008 / DAP008) projektiert werden.

In der IO-Konfiguration der ALU320 wird die MSK351 mit dem Identcode 80 und jeweils 4 Inputbyte und 4 Outputbyte projektiert. Die Input/Outputbyte der ALU entsprechen den Datenbyte 1-4 wie in der folgenden Tabelle beschrieben.

Steckplatz	Projektiert als	Bezeichnung
2	DEP008	Statusbyte1 von MSK351 Hardware (DPM Status)
3	DEP008	Statusbyte2 von DEA300 oder ALU320 (Steckplatz der MSK und AWL Status DEA)
4	DEP008	Datenbyte 1 von DEA300 oder ALU320
5	DEP008	Datenbyte 2 von DEA300 oder ALU320
6	DEP008	Datenbyte 3 von DEA300 oder ALU320
7	DEP008	Datenbyte 4 von DEA300 oder ALU320
8	DAP008	Statusbyte3 zur DEA300 oder ALU320 (AWL Status der MSK351)
9	DAP008	Datenbyte 1 zur DEA300 oder ALU320
10	DAP008	Datenbyte 2 zur DEA300 oder ALU320
11	DAP008	Datenbyte 3 zur DEA300 oder ALU320
12	DAP008	Datenbyte 4 zur DEA300 oder ALU320

4.1.1 Statusbyte1 von MSK351 Hardware (Steckplatz 2)

Bit 0	PAB Signal 5V vorhanden (Bereit)
Bit 1	PAB Signal USWN ist OK
Bit 2	PAB Signal TOTUSN ist OK
Bit 3 – 5	Reserviert
Bit 6	PAB Ready (USWN und TOTUSN sind OK)
Bit 7	DPM Hardwareankopplung DEA300 (ALU320) ← → MSK351 erfolgreich

4.1.2 Statusbyte2 von DEA300 oder ALU320 (Steckplatz 3)

Bit 0 – 4: Steckplatz der MSK351
Bit 5: Reserviert
Bit 6: Reserviert
Bit 7: AWP Status der DEA300 bzw. ALU320 (0= AWPsteht, 1= AWP läuft)

4.1.3 Statusbyte3 zur DEA300 oder ALU320 (Steckplatz 8)

Bit 0 – 6: Frei für Anwenderprogramm
Bit 7: AWP Status der MSK351 (0= AWP steht, 1= AWP läuft)

Der AWP Status muss auf der MSK351 durch das Anwenderprogramm (Multiprog) erzeugt werden!

PABmaster: Der PAB A320 (DEA300) Treiber liest dieses Register und schreibt den Wert in das 2. Systemmerker-Statusbyte des MSK Steckplatzes.

PABslave: Die MSK351 beschreibt dieses Register über die erste DAP8 Karte (Steckplatz 8).
Es dient zur Zustandsbestimmung des MSK351 AWP.

4.2 Physikalische Ein- / Ausgänge

Die 16 physikalischen Eingänge der MSK351 werden auf den DEP008 der Steckplätze **13** und **14** abgebildet. Die 12 physikalischen Ausgänge befinden sich auf den Steckplätzen **15** und **16**.

Auf den Eingangswörtern des Steckplatzes 17 befindet sich der analoge Eingang. Auf dem ersten Eingangswort liegt dabei der Rohwert des analogen Eingangs, auf dem dritten Eingangswort der kalibrierte Wert. Dieser ist auch für die Weiterverarbeitung der Daten zu benutzen.

Steckplatz	Projektiert als	Bezeichnung
13	DEP008	Physikalische Eingänge 1 – 8 (Klemmen 1 – 8)
14	DEP008	Physikalische Eingänge 9 – 16 (Klemmen 9 – 16)
15	DAP008	Physikalische Ausgänge 1 – 8 (Klemme 17 – 24)
16	DAP004	Physikalische Ausgänge 9 – 12 (Klemme 25 – 28)
17	ADU002	Wort 1 = Rohwert des analogen Eingangs (0 … 20 mA) Wort 2 = Kalibrierter Wert des analogen Eingangs (0… 20mA = 0…4095) Wort 3 und 4 nicht verwendet



4.3 Modbus-Registeradressen im DEA300-Magazin

Wird die MSK351 über eine DEA300 zum Modbus TCP-Client gekoppelt sind die Modbus-Register wie folgt zugeordnet:

Modbus-Zugriff durch Client	MSK351 Pseudo-Baugruppe	Steckplatz MSK351 im DEA-Magazin																			Byte
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Write Register	Slot 4	300	302	304	306	308	310	312	314	316	318	320	322	324	326	328	330	332	334	336	LB
	Slot 5	300	302	304	306	308	310	312	314	316	318	320	322	324	326	328	330	332	334	336	HB
	Slot 6	301	303	305	307	309	311	313	315	317	319	321	323	325	327	329	331	333	335	337	LB
	Slot 7	301	303	305	307	309	311	313	315	317	319	321	323	325	327	329	331	333	335	337	HB
Read Register	Slot 8	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	HB
	Slot 9	56	58	60	62	64	66	68	70	72	74	76	78	80	82	84	86	88	90	92	LB
	Slot 10	56	58	60	62	64	66	68	70	72	74	76	78	80	82	84	86	88	90	92	HB
	Slot 11	57	59	61	63	65	67	69	71	73	75	77	79	81	83	85	87	89	91	93	LB
	Slot 12	57	59	61	63	65	67	69	71	73	75	77	79	81	83	85	87	89	91	93	HB

5 Projektierung der Initialisierungsdateien

5.1 Einstellungen in der ALU020.INI

Im ROOT-Verzeichnis der CF-Karte befindet sich die Datei ALU020.INI. Diese Datei beinhaltet die Einstellungen bezüglich der IP-Adresse und diverser Dienste für den Zugriff auf die ALU020.

```
#ALU020:  (GEN)

[IP]
ADDRESS             = 192.168.0.32
NETMASK             = 255.255.255.0
DHCP                = 0

[FTP]
ENABLE              = 1

[TELNET]
ENABLE              = 1

[TIME-GPS-NMEA]
ENABLE              = FALSE
PORT                = 2
BAUD                = 4800              # darf nicht geändert werden
REFRESH             = 60                #  in Minuten
```

5.1.1 Einstellen der IP-Adresse auf der MSK

ADRESSE und NETMASK müssen immer vorgegeben werden. Wenn Sie die MSK über ein Gateway betreiben möchten, können Sie die Datei wie folgt erweitern.

Beispiel:

ADDRESS= 192.168.101.101
NETMASK= 255.255.255.0
GATEWAY= 192.168.101.1

Sofern Sie die Datei über einen FTP Zugriff geändert haben, müssen Sie die MSK Aus- und Einschalten, damit die neuen Daten übernommen werden.

5.1.2 Zugriff über FTP oder TELNET

Die Einträge [FTP] und [TELNET] ermöglichen den Zugriff auf die MSK mit den entsprechenden Diensten. Wenn Sie den Zugriff über diese Dienste nicht erlauben wollen, setzten Sie ENABLE =0.

Achtung: Wenn Sie den FTP-Zugriff deaktivieren, können Sie die INI-Dateien auf der MSK nur noch über einen Card-Reader ändern.

5.1.3 Zeitsynchronisation über GPS

Parameter unter [TIME-GPS-NMEA]

Wird der Parameter **ENABLE** auf TRUE gesetzt, kann die Zeitsynchronisation der MSK über einen GPS-Empfänger erfolgen.

PORT gibt die COM-Schnittstelle (COM 1) an, an den der Empfänger angeschlossen wird.

BAUD gibt die Übertragungsgeschwindigkeit an, mit der der Empfänger die Daten an die MSK überträgt. Dieser Eintrag darf nicht verändert werden.

REFRESH gibt das Zeitintervall in Minuten (1…60) an, in dem die Systemuhr durch den GPS-Empfänger synchronisiert wird. D.h. innerhalb dieser Zeit treffen zwar mehrere Zeitinformationen vom GPS-Empfänger ein, jedoch wird die Systemuhr erst nach Ablauf dieser Zeit durch das nächste gültige Zeittelegramm synchronisiert. Nach jeder erfolgreichen Synchronisation wird der Refresh-Timer wieder neu gestartet. Siehe auch Kapitel 6.2.



5.2 Projektierung des Shared-Memory-Bereichs für Modbus RS232-Slave / TCP-Server

Die im Folgenden beschriebenen INI-Dateien können über die Funktion "ALU-Parameter – Interface-einstellungen bearbeiten" erstellt werden. Die Funktion wird durch Rechtsklick auf die ALU020 in der Bestü-ckungsliste der OHP-Toolbar aufgerufen.

Die MSK351 stellt insgesamt einen Bereich von 16 kB (MB3.0000 bis MB3.16383) für das Shared-Memory zur Verfügung. Dieser Bereich oder Teile davon, können für die Modbus RS232-Slave oder TCP-Server Schnittstelle reserviert werden.

Die Definition des Bereichs erfolgt über INI-Dateien, die auf der Micro-SD Karte abgelegt werden. Diese Dateien PCOSA020.INI und MBSRV.INI befinden sich im Verzeichnis PCOS_OHP.



Weiterführende Informationen siehe Kap. 5.3 und 5.5.

5.3 Freigabe der Service-Routinen und Interfaces

Die einzelnen Service-Routinen und Schnittstellen werden über die Datei PCOSA020.INI frei gegeben. Die Datei befindet sich im Verzeichnis PCOS_OHP auf der Micro-SD Karte.

Bei Auslieferung der Micro-SD Karte sind alle Einträge inaktiv. Durch Entfernen des Doppelschrägstrichs // wird ein Eintrag freigegeben.

Achtung: Schalten Sie in Ihrem Projekt nur die Service-Routinen und Schnittstellen (Interface) frei, die Sie tatsächlich benötigen. Anhand der Einstellungen in dieser Datei werden beim Hochlauf der MSK die Treiberdateien (DLLs) in den Speicher geladen. Jeder freigegebene Treiber belegt Speicherplatz, auch dann, wenn er nicht verwendet wird.

Wenn Sie die Kommunikatiosprotokolle IEC 60870-5-101 oder -104 über den SPS-Konfigurator projektiert haben, wird die Datei PCOSA020.INI automatisch mit den entsprechenden Einträgen erzeugt und kann beim Senden des Projekts zur MSK mit übertragen werden.

<table>
<tr><td colspan="2">#===
#
#	ALU020 PCOS Einstellungen
#	ALU020 PCOS settings
#
#===</td></tr>
<tr><td>[PCOS-SYSTEM]
SharedMemoryAddr=1000
SharedMemorySize=600
TimeSyncDelay=0</td><td>#Speicherbereich für Shared-Memory
#Startadresse ProConOs %MB3.1000
#Anzahl Bytes in Summe bis %MB3.1599
#Gangreserve in Stunden (1…72), 0 = aus</td></tr>
<tr><td colspan="2">#===
#
#	MODBUS Server Einstellungen: Speicher, Schnittstellen
#	max. 4 Schnittstellen möglich
#
#	MMODBUS server settings: memory, interface
#	max. 4 interface are possible
#
#===</td></tr>
<tr><td>[MODBUS-SERVER]

//SharedMemoryOffset=200
//SharedMemorySize=308

//Interface1= SSERIAL
//Interface2= STCP_IP</td><td>#Speicherbereich für Modbus

#Startadresse
#Anzahl Bytes in Summe

#Slave Serial
#Modbus TCP-Server</td></tr>
<tr><td colspan="2">
#===</td></tr>
</table>

<table>
<tr><td colspan="2">#
Service-Routinen für IEC 60870-5-101 oder IEC 60870-5-104
#
#==</td></tr>
<tr><td>[SERVICE]
//Service1= SVC104SA
//Service2= SVC101SA</td><td>#Protokoll IEC 60870-5-104 (TCP/IP)
#Protokoll IEC 60870-5-101 (seriell)</td></tr>
</table>

Beispiel:

Mit der auf der Micro-SD Karte enthaltenen Beispielprojektierung kann durch Freigabe von Interface1 und Service2 z.B. eine Datenübertragung nach IEC 60870-5-101 über die COM1 realisiert werden.

5.3.1 Definition von Shared Memory

Die Startadresse des Shared Memory kann bei 0 (%MB3.000) beginnen. Bitte achten Sie darauf als Startadresse immer gerade Adressen zu verwenden (0, 2, 4, 200, 1000 etc.)

Die Länge kann bis maximal 16384 vorgegeben werden. Geben Sie auch bei der Länge eine gerade Zahl ein.

Im Beispiel steht dem Anwenderprogramm (AWP) ein Shared Memory Bereich von %MB3.1000 bis %MB3.1599 zur Verfügung.

Der gesamte Shared Memory Bereich oder Teile davon können für eine Modbus-Ankopplung verwendet werden.

5.3.2 Modbus RS232

Werden die Einträge für den Modbus durch Löschen der // freigegeben, kann im Beispiel der Bereich %MB3.1200 bis %MB3.1507 über die Modbus RS232-Schnittstelle beschrieben und gelesen werden.

[MODBUS-SERVER]

SharedMemoryOffset=200
SharedMemorySize=308

Interface1= **SSERIAL**

Sie können auch den kompletten Shared Memory Bereich für die Modbus-Übertragung zur Verfügung stellen. Im Beispiel wäre das mit folgenden Einstellungen möglich:

[MODBUS-SERVER]

SharedMemoryOffset=0
SharedMemorySize=600

Interface1= **SSERIAL**

Über die Datei MBSRV.INI wird dieser Bereich auf die einzelne Coils und Register zugewiesen.

Weiterführende Informationen siehe Kapitel 5.5.



5.3.3 Modbus TCP-Server

Werden die Einträge für den Modbus TCP-Server durch Löschen der // beim Interface2 freigegeben, kann im Beispiel der Bereich %MB3.1200 bis %MB3.1507 über Modbus TCP beschrieben und gelesen werden.

[MODBUS-SERVER]

SharedMemoryOffset=200
SharedMemorySize=308

Interface2= **STCP_IP**

Sie können auch den kompletten Shared Memory Bereich für die Modbus-Übertragung zur Verfügung stellen. Im Beispeil wäre das mit folgenden Einstellungen möglich:

[MODBUS-SERVER]

SharedMemoryOffset=0
SharedMemorySize=600

Interface2= **STCP_IP**

Über die Datei MBSRV.INI wird dieser Bereich auf die einzelne Coils und Register zugewiesen.

Weiterführende Informationen siehe Kapitel 5.5.

5.3.4 Service-Routinen

Soll eine Kommunikation über IEC 60870-5-101 und/oder IEC 60870-5-104 aufgebaut werden, sind die entsprechenden Service-Routinen freizugeben. Es ist möglich mehrere Routinen gleichzeitig zu starten.

Die Einstellung

Service1= SVC104SA (zugehörige Parameter-Dateien SVCCOM1.INI und SVCRTU1.INI)

Service2= SVC101SA (zugehörige Parameter-Dateien SVCCOM2.INI und SVCRTU2.INI)

ermöglicht die gleichzeitige Übertragung von Daten über TCP/IP und eine serielle Ankopplung.

5.4 IEC 60870-5-101, -103 oder -104 Kommunikation

5.4.1 Allgemeines

Für die einzelnen Übertragungsprotokolle können auf der MSK bis zu 8 Service-Routinen gestartet werden. Jeder Service-Routine werden Parameterdateien für die Kommunikation (SVCCOM?.INI) und die Definition der PV-Verarbeitung (SVCRTU?.INI) zugeordnet.

Es sind die Protokolle IEC 60870-5-101, IEC 60870-5-103 und IEC 60870-5-104 mit Slave-Funktionalität verfügbar.

Die Parameterdateien SVCCOM?.INI und SVC_RTU?.INI befinden sich im Verzeichnis PCOS-OHP auf der Micro-SD Karte.

Anmerkung: Die einzelnen Parameter der auf der Micro-SD Card enthaltenen Dateien können von dieser Darstellung abweichende Zuweisungen haben.

Ändern Sie die Einträge entsprechend Ihrer Aufgabenstellung und speichern Sie die Dateien anschließend wieder auf der Micro-SD Card.



Hinweis: Wird einem Eintrag ein Doppel-Schrägstrich "//" vorangestellt, wird die entsprechende Zeile nicht interpretiert. Dies kommt einem Löschen der Zeile gleich und der entsprechende Parameter ist nicht aktiviert. Nicht benötigte Parameter können gelöscht oder deaktiviert werden.
Die Defaulteinstellung für gelöschte oder deaktivierte Parameter ist 0, FALSE oder OFF.

Der Eintrag TRUE entspricht ON, der Eintrag FALSE entspricht OFF. Es können jeweils beide Varianten der Bezeichnungen verwendet werden.



Hinweis: Es gilt folgende Maximalanzahl für die zu übertragenden Objekte.

IEC101
Maximal 512 Objekte in Melde- und Befehlsrichtung in der SVCRTUn.INI Datei.
Maximal 128 Byte in Befehlsrichtung und 256 Byte in Melderichtung für die I/O-Konfiguration.

IEC103
Maximal 512 Objekte in Melde- und Befehlsrichtung in der SVCRTUn.INI Datei.
Maximal 128 Byte in Befehlsrichtung und 256 Byte in Melderichtung für die I/O-Konfiguration.

IEC104
Maximal 1024 Objekte für Service 1 und 2 und max. 512 Objekte für alle weiteren Service-Routinen in der SVCRTUn.INI Datei.
Maximal 128 Byte in Befehlsrichtung und 256 Byte in Melderichtung für die I/O-Konfiguration.



Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-104 (SVCCOMn)

[IEC-5-104]	
PORT= 2404	Portnummer **Darf nicht geändert werden!**
KVAL= 12	Maximale Differenz zwischen Anzahl Empfangsfolgen und Sendefolgen
WVAL= 8	Späteste Quittierung nach Empfang von w APDU im I-Format
T1= 15	Zeitüberwachung für gesendete APDU oder Test-APDU
T2= 10	Zeitüberwachung für Quittierungen, falls keine Daten übertragen werden
T3= 40	Zeitüberwachung für gesendete Testframes
IP_ADDR=192.168.0.1	IP-Adresse der Gegenstelle

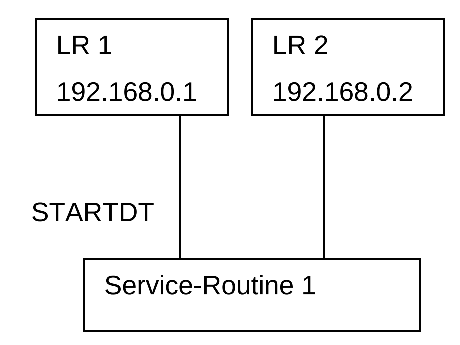
Über eine Service-Routine können maximal zwei IEC 60870-5-104 Verbindungen aufgebaut werden. Jedoch darf nur bei **einer** Verbindung das STARTDT aktiviert werden. Bei der 2. Verbindung werden nur die Testframes (TESTFR) bestätigt.

Sie können die IP-Adresse des Kommunikationspartners eintragen (max. 2). Damit wird sichergestellt, dass die MSK eine TCP/IP-Verbindung nur zu dieser Adresse zulässt. Verbindungsversuche von anderen Adressen werden nicht bestätigt.

Wird keine Adresse eingetragen, kann über jede beliebige IP-Adresse im gleichen Subnetz eine Verbindung etabliert werden.



Hinweis: Werden für eine redundante Datenübertragung zwei Verbindungen, mit aktiviertem STARTDT benötigt, müssen zwei Service-Routinen projektiert werden.



LR 1 192.168.0.1

LR 2 192.168.0.2

STARTDT

STARTDT

Service-Routine 1

Service-Routine 2

Daten werden nur mit LR1 ausgetauscht.

Bei LR2 werden nur TESTFR beantwortet.

Ein Wechsel kann nur durch STOPDT bei LR1 und anschließendem STARTDT bei LR2 veranlasst werden

Daten werden mit beiden LR ausgetauscht

5.4.2 Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-101 (SVCCOM?.ini)

Öffnen Sie die Datei SVCCOM?.INI die Sie auf Ihrem Rechner gespeichert haben mit dem Editor.
Sie sehen jetzt folgende Einträge:

[IEC-5-101]		
AWD= FALSE	TRUE = AWD-Betrieb, FALSE = Standleitungsbetrieb	
PORT= 1	Port =1 oder 2 entsprich COM - Schnittstelle.	
BAUDRATE= 9600	Übertragungsrate zum Leitrechner	
PARITY= EVEN	Parität NO / EVEN	(bei Standleitung immer EVEN)
DATA=8	Anzahl Datenbits	**Darf nicht geändert werden!**
STOPBIT=1	Stoppbit	**Darf nicht geändert werden!**
TVS2=25	Vorlaufzeit S2 [ms]	(nur relevant bei Standleitung)
TNS2=7	Nachlaufzeit S2 [ms]	(nur relevant bei Standleitung)
TVM2=40	Vorlaufüberwachung M2 [ms]	(nur relevant bei Standleitung)
TNM2=150	Nachlaufüberwachung M2 [ms]	(nur relevant bei Standleitung)
PAUSE=27	Pausenzeit [ms]	
BUS_TOUT= 2	Busüberwachung (Slave) in n*[100ms]	
POLL_TOUT= 60	Überwachung Pollzyklus (Slave) in n*[100ms]	
S2SIGNAL= OFF	S2-Signal (RTS) ein- oder ausschalten	
M2SIGNAL= OFF	M5/M2-Überwachung (DCD/CTS) ein- oder ausschalten	
CONF_SC= OFF	ON= Empfangsbestätigung mit Einzelzeichen E5, OFF = Kurztelegramm	
AVS= 3	Adresse der Verbindungsschicht	
OKTETT_AVS= 2	Anzahl Oktett der AVS (1 bis 2)	
[AWD101-MODE]	!! AT Befehle ASCII, max. 8 Zeichen !!	
AT_INIT= ATZ0	Abgespeichertes Profil in Modem laden	
AT_HANGUP= ATH	AT-Kommando für Auflegen	
AT_DIAL= ATD	AT-Kommando für Wählen	
AT_PICK_UP= ATA	AT-Kommando für Abheben	
# Modem RESPONSE/Rückmeldung Dezimal		

//RSP_OK= 0
//RSP_CONNECT= 1
//RSP_RING= 2
//RSP_NOCARRIER= 3
//RSP_ERROR= 4
//RSP_NODIAL= 5
//RSP_BUSY= 6
	Hier sind die bei den Modems üblichen Werte für die Rückmeldungen angegeben. Sollte das von Ihnen verwendete Modem davon abweichende Werte verwenden, löschen Sie vor dem entsprechenden Eintrag den Doppel-Slash "//" und geben Sie den verwendeten Wert an.	



//RSP_NOANSWER=7
RSP_TOUT= 2
	Maximale Wartezeit auf die Bestätigung der AT-Kommandos (1 bis 15 Sek.)

# Wahlparameter / Verbindungsaufbau	
DIAL_NUMBER= ????????? DIAL_RETRY= 3 DIAL_TM_REDIAL= 30 DIAL_TM_RIDLE= 1 DIAL_TM_CON= 60 DIAL_TM_PW= 2 DIAL_PRI_PW= ??????? DIAL_SEK_PW= ???????	Telefonnummer der Zentrale Wahlwiederholung (1 bis 7) Zeit[Sek]: Pause bei Wahlwiederholung (1 bis 255) Zeit[Min]: Sperrzeit nach Wahlwiederholung (1 bis 255) Zeit[Sek]:Wartezeit auf Verbindung "Connect" (1 bis 255) Zeit[Sek]:Wartezeit bis Passwort nach Verbindungsaufbau gesendet wird Passwort Primärstation (Leitrechner) maximal 16 Zeichen Passwort Sekundärstation (Unterstation) maximal 16 Zeichen

Anmerkung: Die einzelnen Parameter können von dieser Darstellung abweichende Zuweisungen haben.

5.4.2.1 Erläuterung zu den einzelnen Parametern im Standleitungs-Betrieb

PORT

Hier geben Sie die COM-Schnittstelle (1 oder 2) an, die Sie für die Kommunikation mit IEC 60870-5-101 verwenden wollen.

BAUDRATE

Die Baudrate kann zwischen 600 und 9600 Baud eingestellt werden.

PARITY, DATA, STOPBIT

Bei Standleitungsbetrieb oder direkter serieller Kopplung ist die PARITY auf EVEN zu stellen. Die Anzahl Datenbit muss immer 8 sein, die Anzahl Stoppbit immer 1.

TVS2, TNS2, TVM2, TNM2, PAUSE, S2SIGNAL, M2SIGNAL

Über diese Parameter werden die Vor- und Nachlaufzeit für S2 und die Überwachungszeiten für das M2-Signal, sowie die Pausenzeit zwischen den Telegrammen eingestellt. Diese Angaben sind vor allem bei Verwendung von Standleitungsmodems (UEM 202 oder 201) erforderlich. Über die Parameter S2SIGNAL, M2SIGNAL kann das Setzen bzw. Überwachen der jeweiligen Signale abgeschaltet werden.

POLL_TOUT

Innerhalb dieser Überwachungszeit muss die Station angepollt werden, ansonsten wird die Verbindung als gestört gemeldet und kann nur durch ein "Initialisieren der Verbindungsschicht" aktiviert werden.

BUS_TOUT

Die Überwachung **BUS_TOUT** gibt die Überwachungszeit beim Linienbetrieb an, innerhalb der die eigene oder eine andere Station an der Linie (BUS) gepollt werden muss. Ist die Station in diesen Timout gelaufen, erwartet sie ein Initialisieren der Verbindungsschicht.
Bei Punkt-zu-Punkt oder AWD-Betrieb darf der **BUS_TOUT** nicht kleiner als der **POLL_TOUT** eingestellt werden.

CONF_SC

Über diesen Parameter geben Sie vor, wie die Unterstation antworten soll, wenn keine Datentelegramme zur Übertragung anstehen. ON= Empfangsbestätigung mit Einzelzeichen E5, OFF = Kurztelegramm (Telegram fester Länge).

AVS, OKTETT_AVS

AVS ist die Adresse der Verbindungsschicht. Mit dieser wird die Station gepollt. Die Adresse kann 1 oder 2 Oktett (Byte) lang sein. Diese Einstellung muss mit allen Teilnehmern im System über die Kompatibilitätsliste festgelegt werden.

5.4.2.2 Erläuterung zu den einzelnen Parametern im AWD-Betrieb

PARITY, DATA, STOPBIT

Bei den meisten Wähl-Modems oder ISDN-TAs ist die Übertragung von 8 Datenbit nur ohne Paritätsbit möglich. Setzen Sie deshalb den Parameter PARITY= NO. Die Einstellung muss mit der Projektierung auf der Gegenseite übereinstimmen.

RSP_TOUT ist die Überwachungszeit, innerhalb der die Modem-Rückmeldungen (CONNECT, OK etc.) eintreffen müssen. Die vorgegebene Standardeinstellung ist für die meisten Modemtypen passend und muss nicht verändert werden.

Kommt eine von der Unterstation eingeleitete Verbindung nicht innerhalb der mit Parameter **DIAL_TM_CON** angegebenen Zeit zustande, wird der Anwahlversuch abgebrochen. Nach Ablauf der Wartezeit **DIAL_TM_REDIAL** wird erneut versucht die Verbindung aufzubauen. Dies wird solange wiederholt, bis die Verbindung zustande kommt oder die eingestellte Anzahl von Wahlwiederholungen **DIAL_RETRY** erreicht ist.

Ist die vorgegebene Anzahl von Anwahlversuchen erreicht ohne dass eine Verbindung zustande kam, wird ein Timer mit der in **DIAL_TM_RIDLE** angegebenen Zeit gestartet. Nach Ablauf dieser Zeit wird die oben beschriebene Prozedur erneut durchlaufen.

Das Passwort für die Zentrale (**DIAL_PRI_PW)** und die Unterstation (**DIAL_SEK_PW**) kann maximal 16 Zeichen lang sein. Wird das Passwort mit weniger Zeichen angegeben, wird es automatisch mit Leerzeichen aufgefüllt. Groß/Kleinschreibung wird geprüft. Die angegebenen Passwörter müssen mit der Projektierung im Leitsystem übereinstimmen.

Nachdem ein Verbindungsaufbau erfolgreich eingeleitet wurde (Connect vom Modem) sendet der Leitrechner bzw. die Unterstation sein/ihr Passwort. Das Passwort wird immer von der Seite gesendet, die den Verbindungsaufbau eingeleitet hat und von der Gegenseite geprüft. Ist die Prüfung erfolgreich, wird der Pollbetrieb aufgenommen. Wird bei der Passwortprüfung ein Fehler erkannt, unterbricht die prüfende Seite die Verbindung mit ATH.



5.4.3 Parameterdatei für die Datenpunktdefinition bearbeiten (SVCRTU?.ini)

[RTU_I10X] ASDU= 3 OKT_ASDU= 2 OKT_HERK=TRUE OKT_AINF= 3	 ASDU = Stationsadresse Anzahl Oktett der Adressbyte Übertragungsursache mit Herkunftsadresse Anzahl Oktett der Informationsobjektnummer
APDU_LEN=253 BLOCK_OBJEKT= TRUE BLOCK_ELEMENT= TRUE	Maximale Länge der Blocktelegramme Blockung von Informationsobjekten erlaubt TRUE / FALSE Blockung von Informationselementen erlaubt TRUE / FALSE
IEC_CCI_TYP= O	Zählwertabfrage Abfrage Betriebsart [C | D | O]
RTU_COM_STOP=TRUE	Verhalten bei STOP des Anwenderprogramms, TRUE= Kommunikation stoppt, FALSE= läuft weiter
EXEC_CMDNENABLE = TRUE EXEC_CMDNTIME = 10 EXEC_STIME= 3 EXEC_LTIME= 50	Befehle mit Qualitätskennung QU=0 sind erlaubt Befehlsausführungszeit für Befehle ohne zusätzliche Festlegung (QU=0) Kurze Befehlsausführungszeit (n * 100 ms) Lange Befehlsausführungszeit (n * 100 ms)
1:AP_NMB= 3600 1:AP_WARN= 2500	Anzahl Telegramme im Archiv für PVs mit Zeitmarke (0 bis 3600) Überlaufwarnung des Archivpuffers (führt bei AWD-Betrieb zum Verbin-dungsaufbau).
OOFS_SP=100 OOFS_SP_TM=100 OOFS_DP=200 OOFS_DP_TM=200 OOFS_BO=300 OOFS_BO_TM=300 OOFS_ME=400 OOFS_ME_TM=400 OOFS_IT=500 OOFS_IT_TM=500	Objektnummern - Offset Einzelmeldungen (Single Point) Objektnummern - Offset Einzelmeldungen (Single Point) mit Zeitmarke Objektnummern - Offset für Doppelmeldungen (Double Point) Objektnummern - Offset für Doppelmeldungen (Double Point) mit Zeitm. Objektnummern - Offset Bitstring Objektnummern - Offset Bitstring mit Zeitmarke Objektnummern - Offset Messwerte Objektnummern - Offset Messwerte mit Zeitmarke Objektnummern - Offset Zählwerte Objektnummern - Offset Zählwerte mit Zeitmarke
OOFS_SC= 600 OOFS_DC=700 OOFS_SE=800 OOFS_CBO=900	Objektnummern – Offset Einzelbefehl (Single Command) Objektnummern - Offset Doppelbefehl (Double Command) Objektnummern - Offset Sollwert normiert (Set-point command) Objektnummern - Offset Bitstring (Bitstring of 32 Bit))

# interne Anweisung für die Varianten (Datenverarbeitung) # Datenvariante Var1	
1:VAR_XX_ABF= TRUE 1:VAR_XX_SPO= TRUE 1:VAR_XX_AWD= FALSE 1:VAR_XX_AP= 1 1:VAR_XX_GRP=1	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Spontan Übertragung freigeben 1= bei AWD Betreib löst Änderung Verbindungsaufbau aus Zyklisch in Archivpuffer 0 = nein, Archiv 1…3 Wert wird bei GA mit Gruppenabfrage 1 übertragen
# interne Anweisung für die Variante1 nur für Doppelmeldungen(DP)	
1:VAR_DP_TS=100	Störstellungsunterdrückungszeit, 0 = keine Überwachung, n * 10 ms
# interne Anweisung für die Variante1 nur für Bitstring(BO)	
1:VAR_BO_NMB=32	Anzahl Prozesspunkte, 1…32 pro Bitstring
# interne Anweisung für die Variante1 nur für Zählwerte(IT)	
1:VAR_IT_IMPS= 0	Impulsschwelle (threshold) 0 bis 65535 (def. = 0, aus)
# interne Anweisung für die Variante1 nur für Zählwerte (IT) und Messwerte (ME)	
1:VAR_ITME_ZYK = 5	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante 1nur für Messwerte (ME)	
1:VAR_ME_AZI=512	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8 12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
1:VAR_ME_OV=32000	Overrange (OV) von 0 bis 32767 (0x7FFF)
# interne Anweisung für die Varianten (Datenverarbeitung) # Datenvariante Var2	
2:VAR_XX_ABF= TRUE 2:VAR_XX_SPO= TRUE 2:VAR_XX_AP= 0 2:VAR_XX_GRP=2	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Spontan Zyklisch in Archivpuffer 0 = nein, Archiv 1…3 Wert wird bei GA mit Gruppenabfrage 2 übertragen
# interne Anweisung für die Variante2 nur für Zählwerte(IT)	
2:VAR_IT_IMPS= 20	Impulsschwelle (threshold) 0 bis 65535 (def. = 0, aus)
# interne Anweisung für die Variante2 nur für Zählwerte (IT) und Messwerte (ME)	
2:VAR_ITME_ZYK = 0	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante2 nur für Messwerte (ME)	
2:VAR_ME_AZI=256	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8 12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
2:VAR_ME_OV=32000	Overrange (OV) von 0 bis 32767 (0x7FFF)



#interne Anweisung für die Varianten (Datenverarbeitung) # Datenvariante Var3	
3:VAR_XX_ABF= TRUE 3:VAR_XX_SPO= FALSE 3:VAR_XX_AP= 1	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Keine spontane Übertragung Zyklisch in Archivpuffer 0 = nein, Archiv 1…3
# interne Anweisung für die Variante3 nur für Zählwerte (IT) und Messwerte (ME)	
3:VAR_ITME_ZYK = 5	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante3 nur für Messwerte (ME)	
3:VAR_ME_AZI=0	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8 12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
	Es können bis zu 16 verschiedene Verarbeitungsvarianten definiert werden.
# Verknüpfung (Zuordnung) von Prozessdaten in Melderichtung (Monitoring Direction)	
QX0.0:TAG_SP_TM= 1;V1	Einzelmeldung mit Zeitstempel
QX0.0:TAG_SP= 1;V1 QX0.1:TAG_SP= 2;V1	Einzelmeldungen ohne Zeitstempel
QX0.4:TAG_DP=1;V1	Doppelmeldung
QD4:TAG_BO=1;V2	Bitstring 32 Bit
QW8:TAG_ME= 1;V1; QW10:TAG_ME= 2;V2;	Messwerte ohne Zeitstempel definiert als WORD
QW12:TAG_ME_TM= 3;V1; QW14:TAG_ME_TM= 4;V2;	Messwerte mit Zeitstempel definiert als WORD
QD16:TAG_IT= 1;V1 QD20:TAG_IT_TM= 2;V1 QD24:TAG_IT= 3;V2 QD28:TAG_IT_TM= 4;V2	32 Bit Zählwerte definiert als DWORD

# Verknüpfung (Zuordnung) von Prozessdaten in Befehlsrichtung (Control Direction)	
IX0.0:TAG_SC= 1;V1 IX0.1:TAG_SC= 2;V1 IW2:TAG_SE=1;V1 IW4:TAG_SE=2;V1	Im 1. Byte des Übergabefels werden zwei Einzelbefehle eingetragen Im 3. und 4. Byte des Übergabefelds wird ein Sollwert eingetragen Im 5. und 6. Byte des Übergabefelds wird ein Sollwert eingetragen

Erläuterung zu den einzelnen Parametern

ASDU, OKT_ASDU:

Die **A**pplication **S**ervice **D**ata **U**nit ist die Stationsadresse. Die Länge ist nur bei IEC 60870-5-101 variabel, bei IEC 60870-5-104 ist sie fest mit 2 vorgegeben.

OKT_HERK:

Gibt an, ob die Übertragungsursache mit oder ohne Herkunftsadresse übertragen wird. Bei IEC 60870-5-104 ist das Oktett für die Herkunftsadresse immer enthalten, wird von der MSK jedoch fest auf NULL gesetzt.

OKT_AINF:

Länge der Adresse des Informationsobjekts. Sie kann 1, 2 oder 3 Byte groß sein. Die Länge ist nur bei IEC 60870-5-101 variabel, bei IEC 60870-5-104 ist sie fest mit 3 vorgegeben.

APDU_LEN= 30

Die maximale Länge der APDU (Protokolldateneinheit der Anwenderschicht = Telegramm) kann vorgegeben werden. Bei IEC-870-5-104 ist die maximale Läge 253 Byte (255 abzüglich Start- und Längenoktett).

BLOCK_OBJEKT

Dieser Schalter gibt an, ob mehrere PVs als geblockte Informationsobjekte übertragen werden dürfen.

Defaulteinstellung = TRUE, da hierdurch eine optimierte Übertragung der PVs insbesondere bei GA möglich ist. Der Schalter sollte nur auf FALSE gestellt werden, wenn das übergeordnete System diese Art von Blocktelegrammen nicht verarbeiten kann.

BLOCK_ELEMENT

Dieser Schalter gibt an, ob mehrere PVs als geblockte Informationselemente übertragen werden dürfen.

Defaulteinstellung = TRUE, da hierdurch eine optimierte Übertragung der PVs insbesondere bei GA möglich ist. Der Schalter sollte nur auf FALSE gestellt werden, wenn das übergeordnete System diese Art von Blocktelegrammen nicht verarbeiten kann.

IEC_CCI_TYP= O

Die Norm sieht für die Übertragung von Zählwerten 4 Verfahren vor. Betriebsart A, B, C und D. Die Betriebsart A kann durch den Parameter VAR_ITME_ZYK bei der Definition der Verarbeitungsvarianten realisiert werden.

Bei den Betriebsarten B, C und D wird ein Zählerabfragebefehl (CI = Counter Interrogation) vom übergeordneten System geschickt. Die MSK unterstützt nur die Betriebsarten C und D, sowie ein OHP spezifisches Verfahren (O) bei dem die Zählwerte sofort nach einem Umspeicherbefehle (Zählerabfragebefehl mit FRZ=1, RQT=1…5) mit der Ursache 37 … 41 übertragen werden. Dies ist eine Abwandlung der Betriebsart C (eine Zählerabfrage FRZ=0, RQT=1…5 wird hier nicht gesendet).



RTU_COM_STOP

Über diesen Schalter wird vorgegeben, ob die Kommunikation zum Übergeordneten System bei AWP-Stopp (AWP = Anwenderprogramm) ebenfalls gestoppt werden soll oder nicht.

FALSE = Kommunikationstreiber laufen weiter und antworten auf Pollaufrufe mit Kurzantwort oder E5. Alle Projektierten PVs werden bei AWP-Stopp mit gesetztem Invalid-Bit übertragen. Wird das AWP wieder gestartet werden alle Invalid-Bits zurückgesetzt und die PVs werden übertragen.

RTU_TIME_SET

0 = Zeit darf über IEC gestellt werden (default)
1 = Zeit darf nie über IEC gestellt werden
2 = Zeit darf nur über IEC gestellt werden, wenn aktuelle Zeit IV (ungültig en: invalid) ist

RTU_TIME_ACK_NEG

FALSE = IEC Zeittelegramm wird mit positiver Response bestätigt (default)
TRUE = IEC Zeittelegramm wird mit negativer Response bestätigt

Dieser Parameter ist nur gültig, wenn RTU_TIME_SET ungleich NULL ist!

RTU_TIME_SYNC_ENABLE

Eingabe in Minuten (1…65535)
Delta-Zeitangabe in Minuten, seit dem letzten Stellen der Uhr. Nach Ablauf darf die Uhr über IEC-Telegramm gestellt werden.
Dieser Parameter ist nur gültig, wenn RTU_TIME_SET = 2 ist!

EXEC_CMDNENABLE, EXEC_CMDNTIME:

Wenn der Parameter EXEC_CMDNENABLE = TRUE gesetzt wird, werden Befehle mit der Qualitätskennung OU = 0 (ohne zusätzliche Festlegung) entgegen genommen und mit der in Parameter EXEC_CMDNTIME festgelegten Zeit ausgegeben. Wird der Parameter EXEC_CMDNENABLE = FALSE gesetzt, werden Befehle mit QU=0 negativ bestätigt und nicht ausgeführt.

Normalerweise wird diese Einstellung verwendet, wenn die Befehlsausführungszeit in der Station für jeden Befehl individuell festgelegt wird. Sollte dies im Anwenderprogramm so realisiert sein, ist zu beachten, dass die MSK die "Beendigung der Aktivierung" (Ursache 10) immer nach Ablauf der durch EXEC_CMDNTIME vorgegebenen Zeit sendet.

EXEC_STIME, EXEC_LTIME:

Hier wird die kurze und lange Befehlsausgabezeit vorgegeben. Mit welcher Zeit der Befehle ausgegeben wird, wird durch die Befehlskennung im Telegramm von der Zentrale festgelegt. Zeitbasis: n * 100 ms.

?:AP_NMB, ?:AP_WARN:

Es wird die Anzahl von Telegrammen die der Archivpuffer fassen soll vorgegeben (AP_NMB).

Der Parameter ?:AP_WARN ist nur bei AWD-Betrieb aktiv. Er gibt an, bei welchem Füllstand des Puffers von der Unterstation automatisch eine Verbindung zum übergeordneten System eingeleitet wird.

Es können maximal 3600 Telegramme gepuffert werden. Diese Anzahl kann auf bis zu 3 Archive verteilt werden.

OOFS_??:

Über diese Parameter können Sie pro Datentyp einen Offset vorgeben, der auf die bei der Definition der Eingänge angegebene Objektnummer aufaddiert wird. Die Verwendung der Offsets ermöglicht es **alle** Objektnummern eines Datentyps in einen anderen Bereich zu verschieben.

Im obigen Beispiel werden die ersten 2 Bit im ersten Übergabebyte als Einzelmeldungen mit den Objektnummern 101 und 102 übertragen da der Parameter OOFS_SP mit 100 angegeben ist.

Bit 4 und 5 werden als Doppelmeldung mit der Objektnummer 201 übertragen.

Byte 4..7 des Übergabeblocks (definiert als DWORD) werden als Bitstring mit der Objektnummer 301 übertragen.

Die 4 Messwerte, die ab Byte 8 (definiert als WORD) übergeben werden, werden jeweils mit und ohne Zeitstempel mit den Objektnummern 401, 402, 403 und 404 übertragen.

Die 4 Zählwerte, die ab Byte 16 (definiert als DWORD) übergeben werden mit den Objektnummern 501, 502, 503 und 504 übertragen.



Hinweis: Es wird empfohlen die Offsets für PVs mit und ohne Zeitstempel gleich einzustellen und die Eingänge fortlaufend durchzunummerieren. Dadurch können bei einer Generalabfrage die Daten optimiert übertragen werden.

Verarbeitungsvarianten:

Es können bis zu 16 unterschiedliche Verarbeitungsvarianten (1:VAR…, 2:VAR… usw.) definiert werden. Diese werden anschließend den einzelnen Eingängen zugeordnet. Es gibt Parameter die für alle Datentypen gültig sind (gekennzeichnet mit XX), andere beziehen sich nur auf bestimmte Datentypen (z.B. VAR_IT_IMPS gültig nur bei Zählwerten).

?:VAR_XX_ABF

TRUE = Prozessvariable wird bei Generalabfrage (global) übertragen.

?: VAR_XX_GRP=1

Über diesen Parameter kann die Generalabfrage als Gruppenabfrage (Gruppe 1…16) ausgeführt werden.

?:VAR_XX_SPO

TRUE = Prozessvariable wird bei Änderung spontan übertragen

?:VAR_XX_AWD

TRUE = Einleiten des Verbindungsaufbau zur Zentrale bei Änderung der Prozessvariablen

?:VAR_XX_AP

Prozessvariable zyklisch ins Archiv Nr. 1, 2 oder 3 schreiben, 0 = inaktiv

?:VAR_BO_NMB

Über diesen Parameter geben Sie an, wie viele hintereinander liegende Bit im Übergabefeld in einem Bitstring zusammengefasst werden. Bereich: 1…32

?:VAR_IT_IMPS

Impulsschwelle im Bereich 0 bis 65535. Bei Erreichen der projektierten Impulsschwelle wird der Zählwert spontan übertragen (Ursache 3). Einstellung 0 bedeutet, dass der Zählwert nur bei Zählerabfrage oder, falls projektiert, zyklisch übertragen wird.



?:VAR_ITME_ZYK

Zeitintervall für die zyklische Übertragung von Zählwerten und Messwerten. Die Werte werden je nach Einstellung von Parameter OFFS_IT / OFF_ME oder OOFS_IT_TM / OOFS_ME_TM mit oder ohne Zeitmarke und der Ursache 3 (spontan) übertragen.

Angabe in Minuten von 1 bis 3600. Einstellung 0 = keine zyklische Übertragung.

?:VAR_ME_AZI

Einstellen des Abweichungszeitintegrals. Die Messwertänderung wird ab dem Zeitpunkt der letzten Übertragung in 1 Sekundenabständen erfasst und vorzeichenrichtig aufaddiert. Erreicht das Ergebnis dieser Addition den eingestellten AZI, wird der Messwert mit der Ursache 3 (spontan) übertragen. Durch dieses Verfahren wird die Messwertübertragung beruhigt. Langsam ansteigende Messwerte werden verzögert übertragen, schnell ansteigende Werte werden früher übertragen.

Bei Einstellung AZI = 0 wird jede Messwertänderung übertragen falls der Parameter VAR_XX_SPO=TRUE ist. Beachten Sie hierbei, dass Messwerteeingänge generell schwanken.

?: VAR_ME_OV

Hier kann die Grenze für den Überlauf (en:Overrange) projektiert werden. Überschreitet der Messwert den hier angegebenen Wert, wird das Überlaufbit in der Qualitätskennung gesetzt. Einstellung 0 = keine Überwachung.

Messwerte werden 1:1 aus dem Übergabefeld übernommen und ins Telegramm eingetragen. Es ist durch das Anwenderprogramm sicher zu stellen, dass die Messwerte im durch die Norm festgelegten Format übergeben werden.

Definition der Übergabeblöcke:

Syntax: Referenz Übergabefeld:IEC-Typ=Objektnummer;Verarbeitungsvariante

Die Referenz auf das Übergabefeld kann als QX?.? (BIT) QB (BYTE), QW (WORD) oder QD (DWORD) angegeben werden. In Befehlsrichtung werden IX?.?, IB, IW und ID verwendet.

Die Referenz ist **relativ** und bezieht sich immer auf das für die Service-Routine definierte Übergabefeld. Wurde in der I/O Konfiguration von MULTIPROG der Output für den Service z.B. von %QB501 bis %QB628 definiert, legt QB0 also die Objektdefinition/Verarbeitung der in %QB501 eingetragenen Daten fest.

Maximale Länge der Übergabeblöcke:

Befehlsrichtung	256 Byte
Melderichtung	512 Byte

Meldungen werden über QX?.? definiert. Bei Doppelmeldungen wird immer das erste der beiden aufeinander folgenden Bit als Referenz angegeben.

Messwerte (16 Bit) werden über QW definiert

Bitstrings und Zählwerte (32 Bit) werden über QD definiert.

Befehle werden über IX?.? definiert wobei bei Doppelbefehlen jeweils das erste Bit der beiden aufeinander folgenden Bit als Referenz angegeben wird.

Sollwerte (16 Bit) werden über IW definiert

Bitstrings (32 Bit) werden über ID definiert.

Beispiele

:

QX0.0:TAG_SP_TM= 1;V1

QX0.0:TAG_SP= 1;V1

Das 1. Bit im Übergabeblock wird als Einzelmeldung (en: **S**ingle **P**oint) sowohl mit, als auch ohne Zeitstempel übertragen. Für die Übertragung gelten die in Variante 1 projektierten Parameter.

QD16:TAG_IT= 1;V1

Der in Byte 16 bis 19 übergebenen Zählwert wird mit den in Verarbeitungsvariante 2 festgelegten Parametern übertragen. Im Beispiel immer nach 5 Minuten und bei Zählerabfrage.

QD28:TAG_IT= 4;V2

Der in Byte 28 bis 31 übergebenen Zählwert wird mit den in Verarbeitungsvariante 2 festgelegten Parametern übertragen. Im Beispiel immer nach 5 Zählimpulsen mit Zeitstempel und bei Zählerabfrage.

TAGs der IEC-Typen:

TAG_SP	Einzelmeldungen (en: **S**ingle **P**oint Information)
TAG_SP_TM	Einzelmeldungen mit Zeitmarke (en: **S**ingle **P**oint Information)
TAG_DP	Doppelmeldungen (en: **D**ouble **P**oint Information)
TAG_DP_TM	Doppelmeldungen mit Zeitmarke (en: **D**ouble **P**oint Information)
TAG_IT	Zählwert (en: **I**ntegrated **T**otal)
TAG_IT_TM	Zählwert mit Zeitmarke (en: **I**ntegrated **T**otal)
TAG_ME	Messwert normiert (**Me**asured Value normalized)
TAG_ME_TM	Messwert normiert mit Zeitmarke (**Me**asured Value normalized)
TAG_BO	Bitmuster von 32 Bit (en: **B**itstring **O**f 32 Bit)
TAG_BO_TM	Bitmuster von 32 Bit mit Zeitmarke (en: **B**itstring **O**f 32 Bit)
TAG_SC	Einzelbefehl (en: **S**ingle **C**ommand)
TAG_DC	Doppelbefehl (en: **D**ouble **C**ommand)
TAG_SE	Sollwert-Stellbefehl normalisiert (en: Set-point command normalized value)
TAG_CBO	Bitmuster von 32 Bit Befehlsrichtung (en: **B**itstring **O**f 32 Bit command direktion)



5.5 Projektierung der Initialisierungsdateien für Modbus RS232-Slave oder TCP-Server

Die Datei MBSRV.INI ist im Verzeichnis PCOS_OHP abgelegt und gilt sowohl für den Modbus RS232-Slave wie auch für den Modbus TCP-Server.

<table>
<tr><td colspan="2">#======================================
MODBUS SERVER / SLAVE Konfigurationsdatei
MODBUS SERVER / SLAVE configuration file
#======================================

#---
Interface 1 - SERIAL - Seriell
#---

Communicationparameter: Modbus Slave Serial Interface - Service 1</td></tr>
<tr><td>[IF1-COM-MODBUS-SSERIAL]
PORT= 2
BAUDRATE= 9600
PARITY= EVEN
DATA=8
STOPBIT=1
PAUSE=10
BUS_TOUT= 2000
ASCII= FALSE
SLAVE_ADR= 1</td><td>
PORT [xx], 0 ... aus

Stoppbit (1= default)
mindesten Pausenzeit [ms] zwischen Request u. Response
Busueberwachung (Slave) in n*[ms], def. (2000), Bereich 1000 bis 2000
Protokoll ASCII FALSE/TRUE; (def.FALSE -> RTU)</td></tr>
<tr><td colspan="2"># Serverparamter für Modbus Slave Serial interface - Service 1</td></tr>
<tr><td>[IF1-SERV-MODBUS]
REF_BASEINDEX= 0
STATUS_OFFSET= 300
STATUS_LEN= 6</td><td># 0/1 default (0)

2, 4 oder 6 Byte lang</td></tr>
<tr><td># 0x COILS - Read/Write - 1 bit access
REF0_OFFSET= 0
REF0_LEN=50</td><td>
REF0: Offset in Byte-Modbus Server
Länge in Byte (50 x 8 COILS)</td></tr>
<tr><td># 1x Discret Input - Read - 1 bit access
REF1_OFFSET= 50
REF1_LEN=50</td><td>
REF1: Offset in Byte-Modbus Server
Länge in Byte (50 x 8 Discrete Inputs)</td></tr>
<tr><td># 3x Input Register - Read - 16 bit access
REF3_OFFSET= 100
REF3_LEN=100</td><td>
REF3: Offset in Byte-Modbus Server
Länge in Byte (100 entsp. 50 Input Register)</td></tr>
<tr><td># 4x Holding Register - Read/Write - 16 bit access
REF4_OFFSET= 20
REF4_LEN=100</td><td>
REF4: Offset in Byte-Modbus Server
Länge in Byte (100 entsp. 50 Holding Register)</td></tr>
<tr><td colspan="2">
#--
Interface 2 - Modbus TCP/IP Server (MODBUS-STCP_IP)
#--

Communicationparameter: Modbus - STCP_IP interface 2</td></tr>
<tr><td>
[IF2-COM-MODBUS-STCP_IP]
//IP_ADDR= 192.168.0.101
//IP_ADDR= 192.168.0.102
//IP_ADDR= 192.168.0.103</td><td>

#IP-Adresse des Teilnehmers
#IP-Adresse des Teilnehmers
#IP-Adresse des Teilnehmers</td></tr>
</table>

//IP_ADDR= 192.168.0.104	#IP-Adresse des Teilnehmers
# Serverparamter für Modbus TCP interface 2	
[IF2-SERV-MODBUS] REF_BASEINDEX= 0 STATUS_OFFSET= 300 STATUS_LEN= 6	# 0/1 default (0) # 2, 4 oder 6 Byte lang
# 0x COILS - Read/Write - 1 bit access REF0_OFFSET= 0 REF0_LEN=50	 # REF0: Offset in Byte-Modbus Server # Länge in Byte (50 x 8 COILS)
# 1x Discret Input - Read - 1 bit access REF1_OFFSET= 50 REF1_LEN=50	 # REF1: Offset in Byte-Modbus Server # Länge in Byte (50 x 8 Discrete Inputs)
# 3x Input Register - Read - 16 bit access REF3_OFFSET= 100 REF3_LEN=100	 # REF3: Offset in Byte-Modbus Server # Länge in Byte (100 entsp. 50 Input Register)
# 4x Holding Register - Read/Write - 16 bit access REF4_OFFSET= 20 REF4_LEN=100	 # REF4: Offset in Byte-Modbus Server # Länge in Byte (100 entsp. 50 Holding Register)

5.5.1 Erläuterung zu den einzelnen Parametern

Mit [IF1…], [IF2…] , [IF3…] usw. wird angegeben auf welches Interface sich die darauf folgenden Parameter beziehen.

[IF1-COM-MODBUS-SSERIAL]

[IF-1-COM-MODBUS-SSERIAL] gibt an, dass diese Einstellungen für das Interface1 mit einer serielle Modbus-Kopplung gelten.

PORT

Hier geben Sie die COM-Schnittstelle (1 oder 2) an, die Sie für die Kommunikation mit Modbus RS232-Slave verwenden wollen.

BAUDRATE

Die Baudrate kann zwischen 600 und 9600 Baud eingestellt werden.

PARITY, DATA, STOPBIT

Bei der Parität kann zwischen EVEN / ODD bzw. GERADE / UNGERADE gewählt werden. Die Anzahl der Datenbit muss immer 8 sein, die Anzahl Stoppbit immer 1.

PAUSE

Pausenzeit in ms zwischen Anfrage und Antwort. Über diesen Parameter kann das Antworttelegramm verzögert werden, falls der Master nicht in der Lage ist, eine Antwort sofort entgegen zu nehmen.

BUS_TOUT

Busüberwachung in ms. Bereich 1000 bis 2000, Defaulteinstellung 2000 (2 Sekunden).

Der Slave prüft, ob innerhalb der eingestellten Überwachungszeit auf dem Bus Telegramme übertragen werden. Ist dies nicht der Fall, wird der Status "Kommunikationsstörung" gesetzt.

ASCII
Schaler für Modbus-ASCII oder Modbus-RTU. Derzeit ist **nur** Modbus-RTU möglich, Einstellung ASCII=FALSE.



SLAVE_ADR
Hier stellen Sie die Slave- bzw. Node- Adresse ein.

[IF1-SERV-MODBUS]

Die folgenden Anweisungen und Parameter gelten für das Interface1 und definieren den Bereich, der über Modbus gelesen und geschrieben werden kann.

REF_BASEINDEX= 0

Hier stellen Sie ein, ob der Modbus-Master das 1. Coil bzw. Register mit 0 oder 1 adressiert.
Defaulteinstellung ist 0.

STATUS_OFFSET, STATUS_LEN

Über diese beiden Parameter wird festgelegt, ab welcher MB3.???? Adresse der Modbus Status abgelegt wird und wie viele Byte für den Status angezeigt werden sollen. Die Länge kann mit 2, 4 oder 6 Byte vorgegeben werden.

Der für die Modbus Kommunikation vorgesehene Shared Memory Bereich kann aufgeteilt werden in Bit- und Wort-Bereiche. Hierfür kann jeweils ein Offset und die Länge des Bereichs angegeben werden. Diese Bereiche können dann mit den entsprechenden Funktionscodes gelesen bzw. beschrieben werden.

In Anlehnung an andere Modbus Projektierungen wurden diese Bereiche wie folgt definiert:

0er Referenz = Ausgangs- oder Merkerbits
1er Referenzen = Eingangsbit
3er Referenzen = Eingangswort
4er Referenzen = Ausgangs- oder Merkerwort

REF0_OFFSET, REF0_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 1	Lese Bit	Read Coils
FC 5	Schreibe einzelnes Bit	Write Single Coil
FC 15	Schreibe Bitbereich	Write Multiple Coils

REF1_OFFSET, REF1_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 2	Lese Eingang	Read Discrete Inputs

REF3_OFFSET, REF3_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 4	Lese Eingangswort	Read Input Register

REF4_OFFSET, REF4_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 3	Lese Register	Read Holding Register
FC 6	Schreibe einzelnes Register	Write Single Register
FC 16	Schreibe Registerbereich	Write Multiple Register
FC 23	Lese/Schreibe Registerbereich	Read/Write Multiple Register

Die Bereiche können auch überlappend projektiert werden.

Mit der Einstellung

REF0_OFFSET= 0
REF0_LEN=50

REF4_OFFSET= 0
REF4_LEN=50

können die ersten 50 Byte des Shared Memory sowohl mit den Funktionscodes FC1, FC5 und FC15 wie auch mit den Funktionscodes FC3, FC6, FC16 und FC23 gelesen bzw. geschrieben werden.

[IF2- COM-MODBUS-STCP_IP]

[IF-2-COM-MODBUS-STCP_IP] gibt an, dass diese Einstellungen für das Interface2 mit einer Modbus TCP-Server Kopplung gelten.

IP_ADDR= ?????
Falls die Modbus TCP-Kopplung nur für bestimmte Clients freigegeben werden soll, können deren IP-Adressen angegeben werden. Es können bis zu 4 Verbindungen gleichzeitig aufgebaut werden. Wird keine IP-Adresse angegeben, erhalten die ersten 4 Clients zugriff.

[IF2-SERV-MODBUS]

Die folgenden Anweisungen und Parameter gelten für das Interface2 und definieren den Bereich, der über Modbus gelesen und geschrieben werden kann.

Beschreibung für Baseindex, Status und Referenzen siehe oben.

Mit der Beispielprojektierung kann der gleiche Speicherbereich mit einer seriellen Modbus wie auch mit einer Modbus TCP/IP Kopplung abgefragt und beschrieben werden, wenn beide Interfaces in der Datei PCOSA020.INI freigegeben werden.

Den beiden Kopplungen können jedoch auch getrennte Bereiche zugewiesen werden. Dies ist dann entsprechen im Shared-Memory und bei der Definition der Referenzen zu berücksichtigen.



6 Tipps und Tricks

6.1 Auslesen und Stellen der MSK Systemzeit

OHP stellt den Anwendern eine Firmwarebibliothek zur Verfügung, die u.a. Funktionen zum Stellen bzw. Auslesen der Systemzeit beinhaltet.

Diese Systemzeit wird nach IEC 60870-5-4, CP56Time2a in 7 Byte wie folgt dargestellt:

<table>
<tr><td>2^7</td><td colspan="6">**Millisekunden**</td><td>2^0</td></tr>
<tr><td>2^{15}</td><td colspan="6">**0…59999**</td><td>2^8</td></tr>
<tr><td>IV</td><td>Res</td><td colspan="6">2^5 Minute 0…59 2^0</td></tr>
<tr><td>S</td><td colspan="2">Res</td><td colspan="5">2^4 Stunden 0…23 2^0</td></tr>
<tr><td colspan="3">2^7 WT 1..7 2^5</td><td colspan="5">2^4 Tag im Monat 1…31 2^0</td></tr>
<tr><td colspan="4">Res</td><td colspan="4">2^3 Monat 1…12 2^0</td></tr>
<tr><td>Res</td><td colspan="7">2^6 Jahre 0…99 2^0</td></tr>
</table>

IV = Ungültig (en: invalid)

IV <0> = gültig
IV <1> = ungültig

S = Sommerzeit

S<0> = Normalzeit
S<1> = Sommerzeit

WT = Wochentag

WT<0> = keine Angabe
WT<1> = Montag
WT<7> = Sonntag

In Multiprog kann diese Zeit z.B. in einem Array oder in einer Struktur abgebildet werden.

```
TYPE
  ArrayTime     :       ARRAY [1..7] OF BYTE;
END_TYPE


TYPE
IEC_Time:
  STRUCT
       ms:          WORD;
       minute:      BYTE;
       hour:        BYTE;
       day:         BYTE;
       mon:         BYTE;
       year:        BYTE;
  END_STRUCT;
END_TYPE
```

Eine Variable vom Typ IEC_Time oder ArrayTime wird an den Funktionsbaustein AX20_ITC_GetCP56Time2a übergeben um die aktuelle Systemzeit auszulesen, bzw. an den Funktionsbaustein AX20_ITC_SetCP56Time2a um die Zeit zu stellen.

6.2 Uhrzeit stellen und MSK-Statusinformationen über Telnet

Über eine Telnet-Verbindung können Sie die MSK-Systemzeit stellen oder den Status der MSK-Systemzeit ermitteln.

- Schließen Sie hierfür den Bedien-PC an die Ethernet-Schnittstelle auf der MSK an.
- Geben Sie im Windows Startmenü im Dialog Ausführen *telnet 192.168.0.20* ein (falls Sie die MSK-IP-Adresse geändert haben, geben Sie die geänderte Adresse ein).

Im Dialogfenster das anschließend erscheint wird die WindowsCE Version und die Firmware-Version angezeigt.

Es stehen Ihnen jetzt folgende Funktionen zur Verfügung:

- Über *time* wird die aktuelle Uhrzeit angezeigt und kann gestellt werden
- Über *date* wird das aktuelle Datum angezeigt und kann gestellt werden
- *DIAG_TSYNC* zeigt den Status der MSK-Systemzeit an

```
Telnet 192.168.0.52
Welcome to the Windows CE Telnet Service on WinCe 4.2 HCA V1.80B1 90608


Pocket CMD v 4.20
\> DIAG_TSYNC

==========================================================
 DIAG_TSync V1.0 - Diagnose time sync.
==========================================================

 --- PCOS I/O services - SVCn ---------------------------
 NO time synchronisation services activ!

 --- SNTP (NTP0:) services -------------------------------
The SNTP services is not available.

 --- GPS services ----------------------------------------
GPS interface COM(1) Baud= 4800 Refresh= 60 minute
Nmb of event: 0
GPS Status  : Error - GPS modul unreachable, no connect.

 --- PCOS time services  --------------------------------
TimeSyncDelay      : OFF
Status time sync. : INVALID
Last time sync from UNKNOWN

 --- Systeminformation ----------------------------------
Operating in the daylight saving mode.
Time Zone         : W. Europe Standard Time
Standard/Daylight: Bias -60/-60 minute
UTC Time          : 04.09.2008 05:48
Local Time        : 04.09.2008 07:48:59:0

\>
```

PCOS I/O services –SVCn

Wurde die Systemzeit über eine Service-Routine (IEC 60870-5-101 oder 104 Uhrzeitsynchronisation) gestellt, wird dies in dieser Rubrik angezeigt.



SNTP <NTP0:> services

Wurde die Systemzeit über einen Zeit-Server gestellt, wir dies hier angezeigt. Die Funktion muss in der A-LU020.INI frei geschaltet werden.

"SNTP service is not available" bedeutet, dass der Service in der ALU020.INI nicht freigegeben wurde,

GPS services

Zeigt den Status der Synchronisation über den Uhrzeitempfänger GPSRVC an. Die Funktion muss in der A-LU020.INI frei geschaltet werden. Siehe Kapitel 5.1.3.

"GPS service is not available" bedeutet, dass der Service in der ALU020.INI nicht freigegeben wurde,

6.3 MSK-Statusinformationen über Telnet

Über eine Telnet-Verbindung können Sie diverse Diagnosefunktionen aufrufen.

- Schließen Sie hierfür den Bedien-PC an die Ethernet-Schnittstelle auf der MSK an oder nutzen Sie eine bestehende GPRS-Verbindung.
- Geben Sie im Windows Startmenü im Dialog Ausführen *telnet 192.168.0. 20* ein (falls Sie die MSK-IP-Adresse geändert haben, geben Sie die geänderte Adresse ein).

In Dialogfenster das anschließend erscheint wird die WindowsCE Version und die Firmware-Version angezeigt.



Hinweis: Die im Folgenden dargestellten Anzeigen sind nur Beispiele und können je nach MSK-Konfiguration von den tatsächlichen Anzeigen abweichen.

services list

Über diesen Aufruf können Sie feststellen, welche Treiber und Routinen auf der MSK gestartet sind.

Beispiel:

```
TEL1:   0x00000000      TELNETD.Dll     Wird ausgeführt
S4U0:   0x00000000      \sd card\pcos_sys\SVC_SEK_I104SUS.dll      Wird ausgeführt
AUC1:   0x00000000      sd card\a020_sys\AX20UDPCFG.dll            Wird ausgeführt
SMB0:   0x00000000      smbserver.dll   Wird ausgeführt
MMQ1:   0x00000000      MSMQD.Dll       Aus
OBX0:   0x00000000      OBEXSrVr.dll    Aus
FTP0:   0x00000000      FTPD.Dll        Wird ausgeführt
TEL0:   0x00000000      TELNETD.Dll     Wird ausgeführt
HTP0:   0x00000000      HTTPD.DLL       Aus
```

diag_cpu

Zeigt ihnen Betriebsystem-, Hardware- und Treiber-Versionen an.

```
============================================
OS Version          : 6.0 Build 0 Pb-Id 3
ALU Version         : 2.50 Build 0.1 14.01.10
PLC SYSTEM          : Micro
PLC ALU             : ALU 020
ProConOs Version    : 4.40
Pcos  Version       : 3.30
--------------------------------------------
MicroSDBoot BuildId:  24022010B01
FPGA HW REVISION   :  1.1
FPGA SW REVISION   :  1.15
DIP_SWITCH         :  0x48
```



Teil III Baugruppenbeschreibung



1 Baugruppenbeschreibung MSK351

1.1 Schnittstellen

Bezeichnung	Steckverbinder	Platzierung
1 Ethernet, Programmierschnittstelle *)	RJ 45	Unterseite
1 Serielle Schnittstellen RS232	D-Sub 9	Unterseite
16 digitale Eingänge, 12 digitale Ausgänge, 1 Analogeingang, Stromversorgung	Schraubklemmen	Frontseite
SD-Steckplatz	Micro SD	Oberseite
1 USB Schnittstelle	Mini USB Buchse	Oberseite
24 VDC zur galvanisch getrennte Einspeisung der E/A (Diese Option muss bei der Bestellung angegeben werden, damit die Baugruppe entsprechend ausgeführt werden kann)	Schraubklemme	Oberseite
PAB Anschlussstecker	A120 Standard	Rückseite

*) **Hinweis:** Bei Auslieferung ist die Ethernet-Schnittstelle auf IP-Adresse 192.168.0.20 voreingestellt.

1.1.1 Schnittstellenbelegung der Schraubklemmen

1.1.1.1 Stromversorgung

Die MSK besitzt ein internes Weitbereichsnetzteil mit einem Eingangsspannungsbereich von 14 VDC – 28VDC. Im Unterspannungsfall < U_{min} wird der Signalspeicher und der Telegrammpuffer der CPU auf der MICRO SD Card nullspannungssicher abgelegt.
Der Anschluss der Versorgungsspannung erfolgt frontseitig über die Klemmen 31 (+) und 32 (Masse). Wird die MSK 351 zusammen mit der ALU320 oder DEA 300 betrieben, so besteht eine galvanische Verbindung zwischen der fronseitig eingespeisten 24V Betriebsspannung und dem PAB Bus der ALU320 oder DEA 300.

Zusätzlich bietet die MSK351 die Möglichkeit, die digitalen E/As galvanisch getrennt zu betreiben. Hierzu ist die 24V Versorgungsspannung zusätzlich an den Klemmen der MSK Oberseite anzuschließen. Die Klemmen an der Geräteoberseite sind doppelt ausgelegt, so dass die 24V hierüber einfach durchgeschleift werden kann.

Achtung: Wenn Sie diese Möglichkeit nutzen wollen, müssen Sie dies bei der Bestellung der MSK351 angeben, damit diese entsprechend ausgeführt werden kann.

1.1.1.2 Schnittstellenbelegung der E/A Schraubklemmen

Frontseitige Belegung der Anschlusselemente

Klemmen	Belegung
1	Eingang 1
2	Eingang 2
3	Eingang 3
4	Eingang 4
5	Eingang 5
6	Eingang 6
7	Eingang 7
8	Eingang 8
9	Eingang 9
10	Eingang 10
11	Eingang 11
12	Eingang 12
13	Eingang 13
14	Eingang 14
15	Eingang 15
16	Eingang 16
17	Ausgang 1
18	Ausgang 2
19	Ausgang 3
20	Ausgang 4
21	Ausgang 5
22	Ausgang 6
23	Ausgang 7
24	Ausgang 8
25	Ausgang 9
26	Ausgang 10
27	Ausgang 11
28	Ausgang 12
29	Stromeingang -
30	Stromeingang +
31	UB 24V
32	GND



1.1.2 PAB-Anschlussstecker

Rückseitig befindet sich der PAB Anschlussstecker nach A120 Standard.

Hinweis: Die MSK351 kann derart ausgeführt werden, so dass sie mechanisch im Rack mit MSK251 betrieben werden kann. Hierzu ist der PAB Stecker der MSK351 nicht montiert. Die Standardausführung ist mit montiertem Stecker.

1.1.3 Serielle RS232 Schnittstelle

Pin-Nr.	COM1 Bez.	Signalbezeichnung
1	DCD	Data Carrier Detect
2	RXD	Receive Data
3	TXD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send

1.1.4 Micro SD Speicherkarten

Die MSK verfügt über einen Speicherkartensteckplatz, für eine Micro-SD-Card. Der Kartensteckplatz befindet sich auf der Oberseite des Gehäuses. Die mitgelieferte Micro-SD-Card hat eine Kapazität von 2 GB.

Die Micro SD Card enthält alle relevanten Daten für die MSK, wie z.B. Betriebs- und Laufzeitsystem, Anwender-applikation und Telegrammpuffer. Die MSK wird mit einer betriebsfertigen Micro-SD-Card ausgeliefert.

Achtung: Die Micro-SD-Card nur im spannungslosen Zustand ziehen und stecken.

1.1.5 Anzeigen

Frontseitige Belegung der Anzeigen

LED Nr.	Bedeutung
1	rot: Eingang 1 aktiv, grün: Eingang 2 aktiv
2	rot: Eingang 3 aktiv, grün: Eingang 4 aktiv
3	rot: Eingang 5 aktiv, grün: Eingang 6 aktiv
4	rot: Eingang 7 aktiv, grün: Eingang 8 aktiv
5	rot: Eingang 9 aktiv, grün: Eingang 10 aktiv
6	rot: Eingang 11 aktiv, grün: Eingang 12 aktiv
7	rot: Eingang 13 aktiv, grün: Eingang 14 aktiv
8	rot: Eingang 15 aktiv, grün: Eingang 16 aktiv
9	rot: Ausgang 1 aktiv, grün: Ausgang 2 aktiv
10	rot: Ausgang 3 aktiv, grün: Ausgang 4 aktiv
11	rot: Ausgang 5 aktiv, grün: Ausgang 6 aktiv
12	rot: Ausgang 7 aktiv, grün: Ausgang 8 aktiv
13	rot: Ausgang 9 aktiv, grün: Ausgang 10 aktiv
14	rot: Ausgang 11 aktiv, grün: Ausgang 12 aktiv
15	grün UB OK
16	grün AWP Run, Stopp, Störung
17	grün reserviert
18	grün reserviert
19	grün reserviert
20	grün reserviert
21	grün reserviert

1.2 MSK Anzeigeelemente

Die MSK enthält 5 LEDs die den Betriebszustand anzeigen.

LED Nr.	Funktion	Farbe	Anzeige
15	UB OK	grün	Dauerlicht
16	AWP Run, Stopp, Störung	grün	Dauerlicht, Dunkel, Blinkt
17	reserviert	grün	
18	reserviert	grün	
19	reserviert	grün	

Währen der Hochlaufphase durchlaufen die LEDs verschieden Blinkzustände.

Nach erfolgreichem Hochlauf geht UB auf Dauerlicht.

Ist ein Anwenderprogramm als Bootprojekt auf der Micro-SD-Card gespeichert, wird dieses gestartet und die LED 16 geht auf Dauerlicht ansonsten ist sie aus.

LED 17, 18 und 19 sind nach dem booten immer aus.

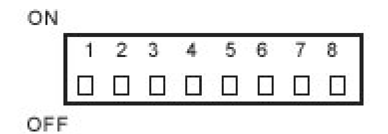


1.3 MSK Bedienelemente

DIL Schalter

Der 8 pol. DIL-Schalter dient zur Einstellung des Startverhaltens der MSK. Der Schalter ist auf der Geräterückseite montiert.

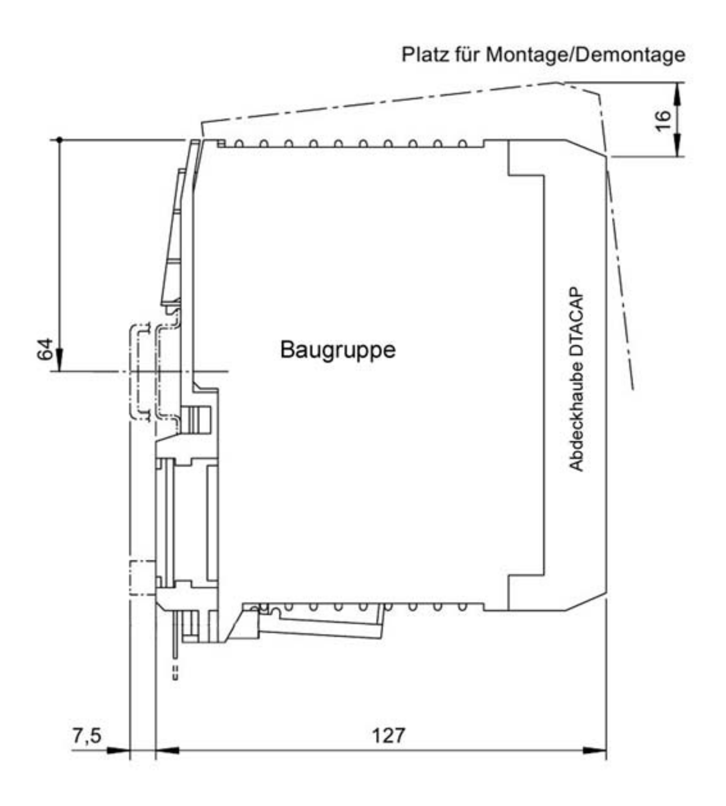
Schalter	Funktion
1	Reserviert
2	Reserviert
3	Reserviert
4	Reserviert
5	**ON = Warmstart, OFF = Kaltstart**
6	Reserviert
7	Reserviert
8	Reserviert





1.4 Montage / Demontage

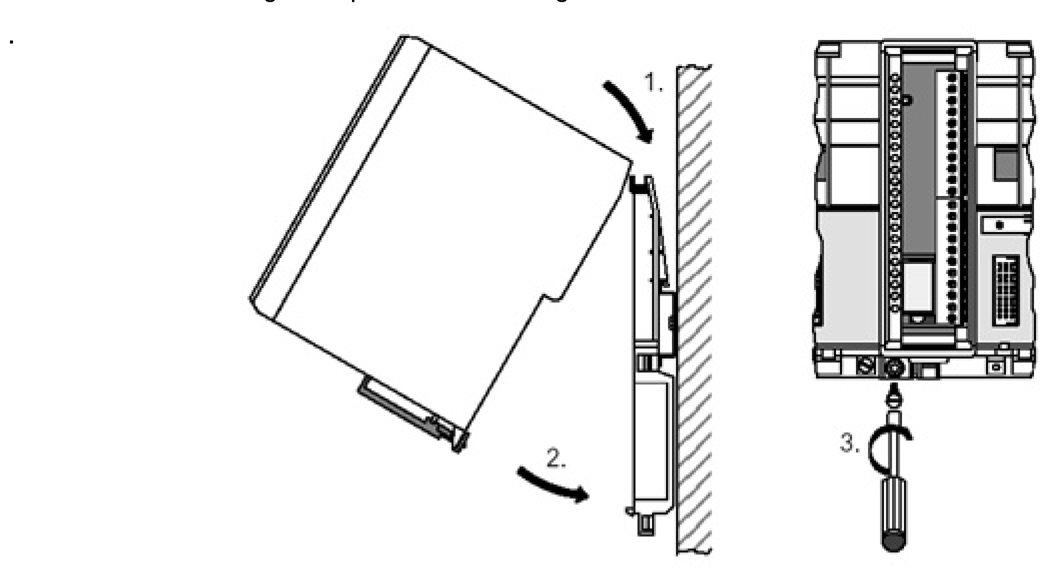
1.4.1 Montageabmessungen



1.4.2 Montage

Führen Sie die Montage entsprechend nachfolgendem Bild durch.

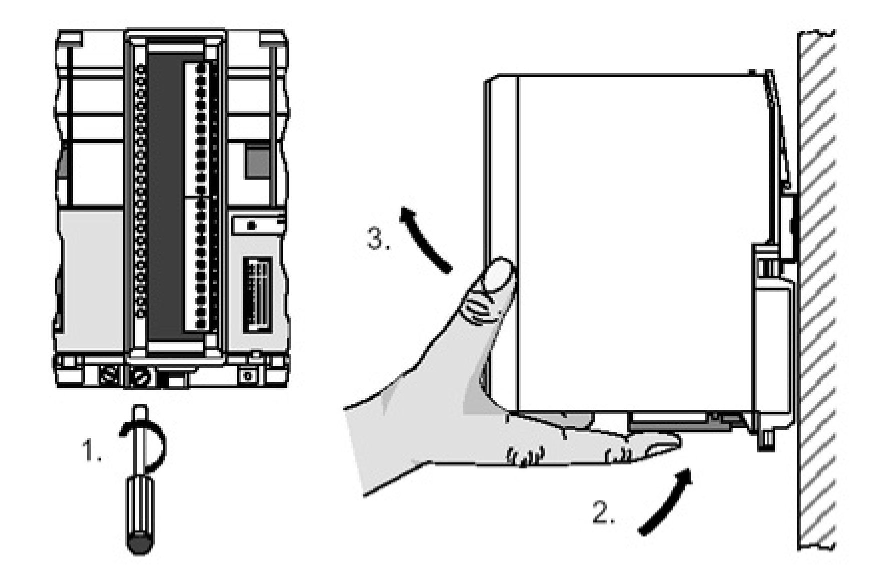
.





1.4.3 Demontage

Ziehen Sie mit dem Ziehgriff die Schraub-/Steckklemmen.
Führen Sie die Demontage entsprechend nachfolgendem Bild durch.



1.5 Technische Daten der MSK

Versorgung	
Versorgungsspannung	14 VDC – 28 VDC
Serielle Schnittstellen	
Anzahl Art, Geschwindigkeit Anschluss	1 seriell, asynchron, V.24 Pegel, max. 19.200 Baud DSUB9, Anschluss Gerätunterseite
Ethernet	
Anzahl Art Anschlussstecker	1 IEEE802.3, 100 Base T, 100 MBit/s RJ45, Anschluss Geräteunterseite
Prozessschnittstellen	
Anzahl Art	2 Frontseitig aufsteckbare 16polige Schraub-/Steckklemmen für Leitungsquerschnitte 0.25 … 1.5 mm^2
Anzahl digitale Eingänge	16 Prozesseingänge für 24V DC, potentialgetrennt mit gemeinsamem Massebezug, angeordnet auf Frontleiste
Anzahl digitale Ausgänge	12 Prozessausgänge für 24V DC 300mA, kurzschluss- und überlastfest, potentialge-trennt
Anzahl analoge Eingänge	1 Analogeingang für Strom, 0 – 20mA, Auflösung 12 Bit, Wandelzeit ca. 10ms, 50 Ohm Impedanz, mit hochohmiger Gleichtaktunterdrückung 0 – 15V
Steckplatz für SD Speicherkarte	
Anzahl Art	1 Standard Compact Flash (Micro SD Card), 2GByte
CPU /Speicher	
Prozessortyp Speicher Telegrammpuffer Betriebssystem	skalierbares CPU Board 32 MByte RAM / 32 MByte Flash 1 MB im Flash, keine Batterie erforderlich Windows CE 6.0 Echtzeitkernel
Uhrzeitbehandlung	
Synchronisation	Über optionales GPS Modul über serielle Schnittstelle
CPU Erweiterungssteckplatz	
Anzahl, Art	1 Steckplatz für Modem Erweiterungskarten
Schutzart	
Schutzart (IEC 60529) Schutzklasse (IEC61140) Isolation	IP40 auf Frontseite IP20 auf allen anderen Seiten I EN60950, IEC 950
Umweltbedingungen	
Umgebungstemperatur bei Betrieb Lagertemperatur Rel. Feuchte (IEC 68-2-1-1/2)	0°C bis 60°C -25°C bis 85°C bis 95% keine Betauung
Anzeigen	
19 LED-Anzeigen	LED 1 – 8: Eingang 1 – 16 (grün + rot) LED 9 – 14: Ausgang 1 – 12 (grün + rot) LED 15: UB OK LED 16: AWL Run, Stopp, Störung LED 17: res. LED 18: res. LED 19: res.